محاضرة ١٥: Sharpening

المحتوى: ① مقدمة

② Derivative Operator

③ Laplacian operator

④ Unsharp Masking and High boost filtering

⑤ Sobel filtering

⑥ Combining spatial Enhancement Methods

مقدمة

* المحاضرة دي فيها رياضيات كتير (مش صعبة)، هحاول أشرحها بشوية تفصيل بس قدر الخص.
* هنعرف الأول ال Derivative operator وبعد كده هنستخدمه في ال Laplacian
* بعدها هنعرف ال Unsharp Masking وازاي نستخدمه في ال Sharpening
* بعدها هنعرف طريقة non-Linear لل Sharpening
* وأخيرا، هنلم اللي عرفناه من طرق ال Enhancement في تطبيق واحد، مش هنطبق بأرقام بس هنشرح آخر سلايد [10.13]

* خدنا ال Smoothing spatial

Smoothing = blurring = Averaging/Integration = Low pass filtering

It was mostly done by linear operation (sum of products like Correlation and Convolution) or non-Linear (median filter)

Sharpening = spatial differentiation = High pass filtering

Can be done by linear operations (Laplacian operator is considered applied by Sum of products) or non-Linear

* أرجع جدا اللي مفهمش الجزء ده يذاكره من المرجع من صفحة 179 لحد صفحة 194

---

* فكرة ال Sharpening هي إن لما يبقى فيه 2 بكسل، والفرق في ال intensity بينهم كبير، فغالبا عشان دول حاجتين مختلفين، وبالتالي عايز يطلع ال edges اللي بينهم ويوضحها أكتر

* ملاحظة مهمة: ال sharpening بيوضح ال noise كمان لو كانت موجودة، و البكسلز اللي فرق ال intensity بينهم مش كبير (طبعا البكسلز المجاورة) بيقلل وضوحها.

* هنستعمل ال first-order derauative وال Second-order derauitive [هنطبقها بشكل Numeric عشان الصورة Digital]

Derauitive Operators

* هنعرف شوية شروط للمشتقة الأولى والثانية لما تبقى discrete، هنشوف المعادلة بتاعتها ونشرح مثال لدالة 1-D، وبعد كده نستخدم الكلام ده في ال 2D image بال Laplacian Operator [أو ال Laplacian filtering]

* راجع سلايد [10.3]، راجع سلايد [10.4] بعد ما تذاكر الجزء ده

* شروط المشتقة الأولى:

① تدي صفر في المناطق اللي ليها Constant intensities

② متديش صفر في بداية ال step وال ramp

③ متديش صفر في ال ramp لما يكون ميلك ثابت

* شروط المشتقة الثانية:

① تدي صفر في المناطق اللي ليها Constant intensity

② متديش صفر في بداية ونهاية ال step وال ramp

③ تدي صفر في ال ramp لما يكون ميله ثابت

[onset = بداية]

حساب المشتقة الأولى والمشتقة الثانية لدالة f(x) (تفاضل جزئي)

$$1st\ order = \frac{\partial f}{\partial x} = \left.\frac{df}{dx}\right|_{x(1D)} = f(x+1) - f(x) \longleftarrow$$

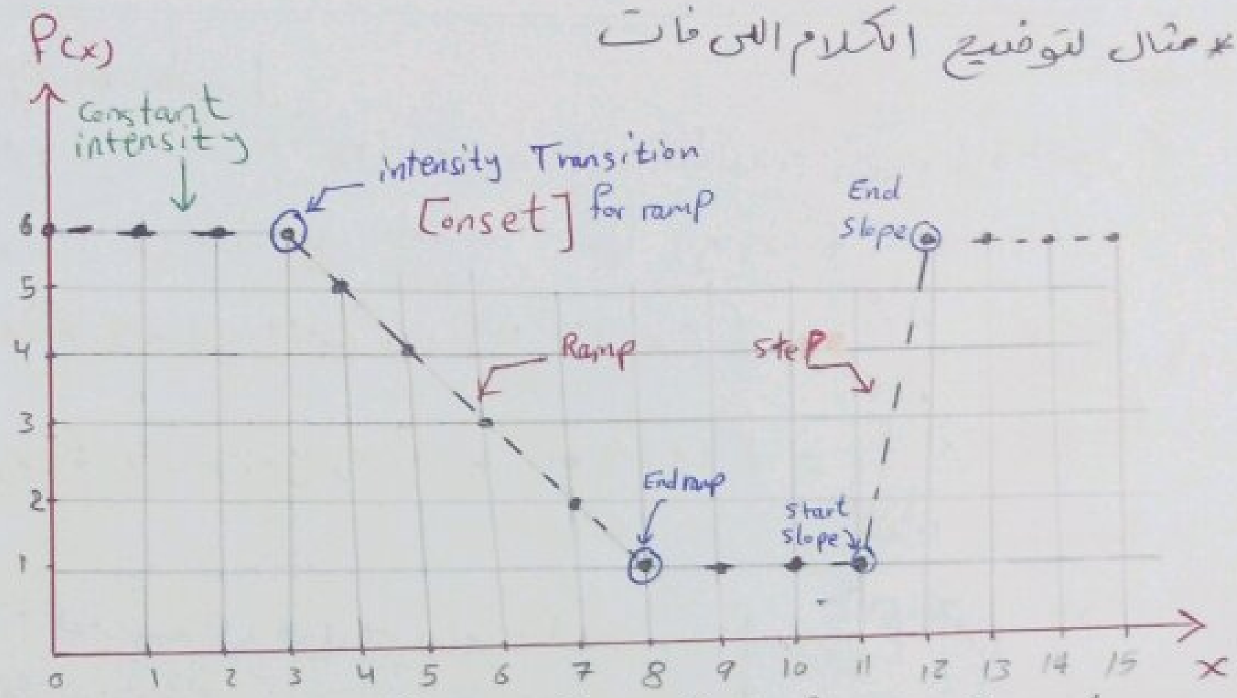
هنسميهم Look-ahead operators واخترنا دالة 1st derivative عشان هنبص قدام

التفاضل الجزئي هو نفس التفاضل الكلي عشان الدالة في x بس

$$2nd\ order = \frac{\partial^2 f}{\partial x^2} = \left.\frac{d^2 f}{dx^2}\right|_{x(1D)} = f(x+1) + f(x-1) - 2f(x)$$

*مثال لتوضيح الكلام اللي فات

Discrete القيم والخط المنقط للتوضيح ال ramp وال step وال Constant



1st Derivative at $x = 6 = f(x+1) - f(x) = f(7) - f(6) = 2 - 3 = -1$

2nd Derivative at $x = 11 = f(x+1) + f(x-1) - 2f(x) = f(12) + f(10) - 2f(11) = 6 + 1 - 2 = 5$

اعمل الجدول بنفس الطريقة

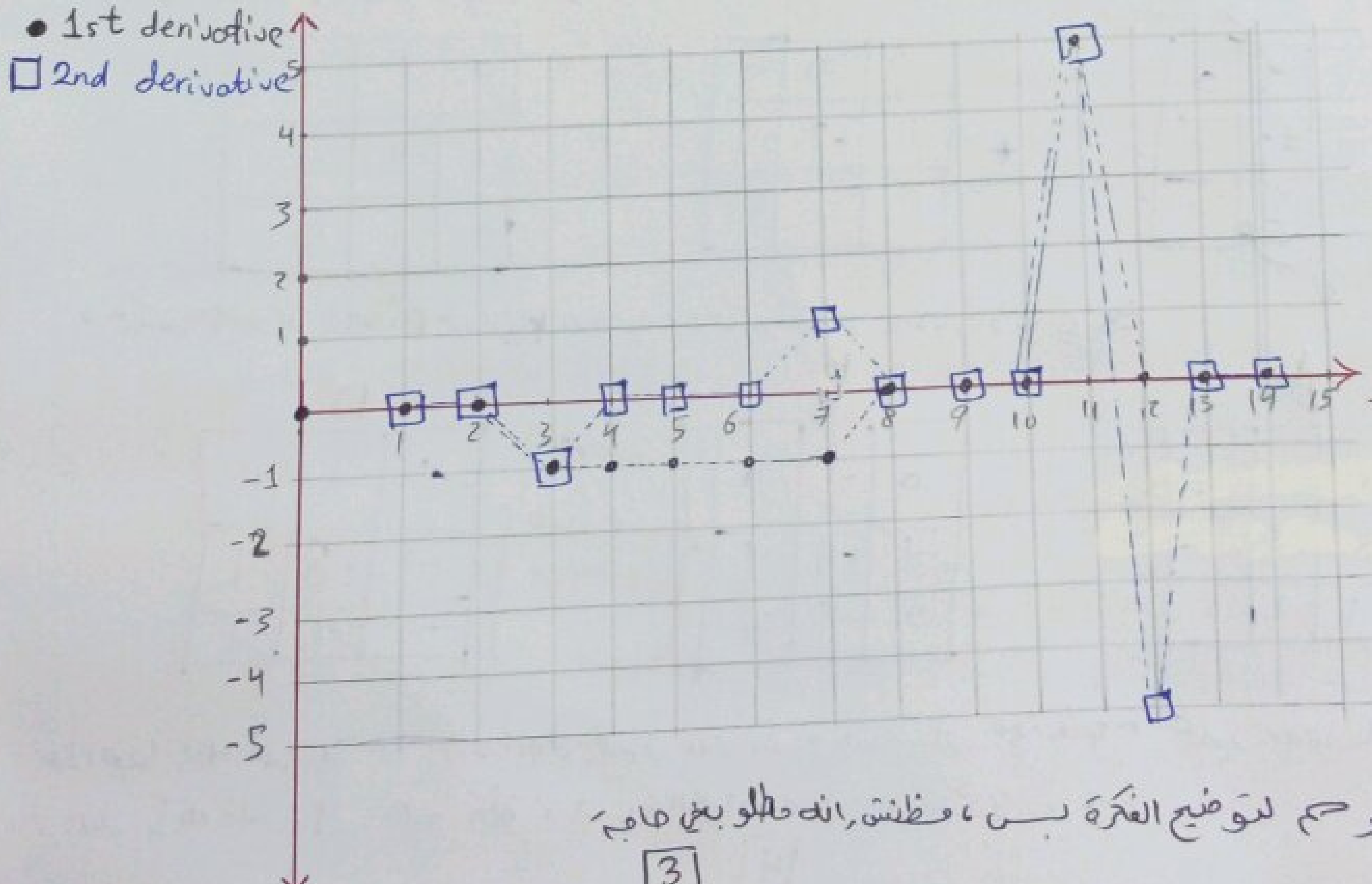
	x=0	x=1	x=2	x=3	x=4	x=5	x=6	x=7	x=8	x=9	x=10	x=11	x=12	x=13	x=14	x=15
1st Derivative	0	0	0	-1	-1	-1	-1	-1	0	0	0	5	0	0	0	
2nd Derivative		0	0	-1	0	0	0	0	1	0	0	5	-5	0	0	

مفيش f(-1)

مفيش f(16)

*هنرسم ال Derivative مع ال x عشان نشوف شكلها عامل ازاي

*راجع الشروط اللي قولناها وشوفنها عالرسم (مهم أوي تفهمها)



الرسم لتوضيح الفكرة بس، مظنش انه مطلوب مني حاجة

Laplacian operator

* هنوظف اللي عرفناه في ال 2D ، هنعمل ال 2nd Derauitve مرة لمحور x ومرة لمحور y وهتكون الصورة عندي اسمها $f(x,y)$

for x: $$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x,y)$$

for y: $$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x,y)$$

هيجمع العمليتين في عملية واحدة اسمها $\nabla^2 f(x,y)\Big|_{(x,y)}$

$$\nabla^2 f(x,y)\Big|_{(x,y)} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

المحاور اللي هتقال عليها → (x,y)

شكل ال Mask (هاخد ال Coefficients من معادلة $\nabla^2 f(x,y)$)

x-mask + y mask = Laplacian Mask (A)

$f(x-1,y)$ ، $f(x,y)$ ، $f(x+1,y)$

x-mask:

0	1	0
0	-2	0
0	1	0

+

y mask:

0	0	0
1	-2	1
0	0	0

=

Laplacian Mask (A):

0	1	0
1	-4	1
0	1	0

* ممكن أدخل ال pixels في الزوايا معايا ، وممكن أعكس الإشارة زي D و C

هدخل البكسلز في الزوايا

B

1	1	1
1	-8	1
1	1	1

C

0	-1	0
-1	+4	-1
0	-1	0

في حالة C, D هتتغير الإشارات في المعادلات

D

-1	-1	-1
-1	+8	-1
-1	-1	-1

ملاحظة: انه عملية ال derivation عملية Linear وبالتالي ال laplacian عملية Linear

* لاحظ ان فلتر y هو فلتر x بس اتعمله rotation ب 90°

تشكل المعادلة $\nabla^2 f(x,y)$ لو خدنا الفلتر (C)

$$\nabla^2 f(x,y) = f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1) + 4\, f(x,y)$$

* لما هطبق ال Laplacian هيطلعلي صورة جديدة ، باجمعها أو أطرحها من الصورة الأصلية
* هطرح لو المعامل ال $f(x,y)$ سالب (ال center بتاع ال Mask) في ال Mask
* هجمع لو معامل ال $f(x,y)$ كان موجب (ال center بتاع ال Mask) في ال Mask
* يبقى لما أجي أعمل Sharpening باستخدام ال Laplacian operator

$$g(x,y) = f(x,y) + c\left[\underbrace{\nabla^2 f(x,y)}\right]$$

الصورة بعد ال Sharpning ← $g(x,y)$

الصورة الأصلية ← $f(x,y)$

c ← موجبة لو ال Center بتاع ال Mask موجب وسالبة لو العكس

الصورة اللي طلعت بعد عمل ال Laplacian Filter

* راجع سلايد [10 . 6]
* عكس ال Laplacian operator طلع قيم سالبة لل pixel intensity ، في الحالة دي هعمل rescale and normalize اللي شرحناها محاضرة 4 سلايد [4 . 10]
* راجع سلايد [10 . 7] ، فيها صور توضح الفرق لما عمل من غير rescale and normalize

(سماها scaling ، اختلاف مسميات كتير بس ما علينا يعني) ولما عمل (scaling)

* ركز كويس جدا في المثال.

Unsharp Masking and Highboost filtering

* دي طريقة تانية عشان يعمل Sharpening
* افتح سلايد [10 . 9] ~~...~~ ، الفكرة سهلة جدا

* راجع سلايد [10.9] كويس لما تخلص الجزئية دي

الفكرة إنه هيستخدم ال blurring عشان يطلع ال edges ، إزاي ؟؟!

- افرض صورة $f(x,y)$ ، وبعد ال blurring بقت $f'(x,y)$ (مجرد مسميات)
- هطلع Mask اسمه $g_{mask}(x,y)$ بطرح الصورة ال blurred من الصورة الأصلية

$$g_{mask} = f(x,y) - f'(x,y)$$

- هضرب ال Mask ده في factor اسمه K (ده هيسمحلي أعلى ال Sharpening شديد ولا لأ ، يعني هل يظهر ال edges أوي ولا لأ)

وأجمع ال Mask على الصورة الأصلية (خلي بالك ال Mask بأبعاد الصورة)

$$g(x,y) = f(x,y) + K g_{mask}(x,y)$$

* عندك مثال على الكلام ده على signal في السلايد [10.9]

* ممكن يبقى فيه قيم بالسالب لل pixels بعد العملية فنعمل Scaling [rescaling & normalizing]

ملاحظة مهمة: ال filters أوقات يقولوا عليها isotropic أو لأ

، معنى إنه الفلتر Isotropic ، إنه هيدي نفس النتيجة لما تعمل rotation لل Mask بتاعك ، زي

Isotropic

0	1	0
1	-4	1
0	1	0

Non-Isotropic

-1	2	-1
0	0	0
1	-2	1

Sobel filtering

* فلاحظ فهمنا إزاي نحسب ال 2nd derivative سواء في ال y أو ال x

وكتبناها على الشكل $\nabla^2 f(x,y)|_{(x,y)}$

* راجع سلايد [10.10] بعد الكلام ده. أو شوف المرجع بداية من صفحة 187 أو سه، وهتفهم الكلام المكتوب.

* ال 1st derivatives ممكن نطبقها من خلال ال Magnitude لل gradient وال gradient هو مجرد vector اللي كنا عملناه قبل كده بس مع اختلاف مسميات، عارف الكلام ده بيلخبط، اعتبر اللي فات تعريف لل 1st derivative وده تعريف تاني في ضوء ال non-linear sharpening باستخدام ال gradient

هنعرف ال gradient بأنه

$$\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

حسبناهم مرة قبل كده كلهم، كل الجديد انهم بقوا قيم جوا vector

* للـ اتجاه، وهنجيب ال Magnitude لل vector ده، وهنسمي ال Magnitude بـ M

$$M(x,y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

$$= \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

* ال M هتبقى صورة بنفس أبعاد الصورة الأصلية لما أمشي على كل البكسلز

* ممكن نعمل Approximation لحساب M ويكون efficient في ال Computation

$$M(x,y) \simeq |g_x| + |g_y| \quad (*)$$

* فيه شوية ملاحظات:

1. ال vector اللي هو ∇f بيشاور على أكبر قيمة للتغير في قيمة f عند بكسل (x, y) وطبعا الـ Magnitude هو قيمة التغير ده (change rate value)
2. عملية ال gradient عملية Linear بس not isotropic
3. ، ال Magnitude = non-linear بس Isotropic
4. الـ Magnitude في حالة المعادلة (*) بيكون non-linear وكذلك non-Isotropic

* ليه الملاحظات اللي فاتت ؟

* عملية ال derivation كانت عملية linear عشان كده ال grad برده linear

* عملية ال Magnitude عبارة عن تربيع للي تحت الجذر أو قيمة مطلقة عشان كده مش linear

* عشان عملية ال grad عملية linear نقدر نحطها في شكل sum of product من خلال ال Masks التالية

المنطقة 3x3 اللي هشتغل عليها في الصورة

$$\begin{array}{ccc} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{array}$$

Roberts Operators

$$x: \begin{array}{|c|c|} \hline -1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} \qquad y: \begin{array}{|c|c|} \hline 0 & -1 \\ \hline 1 & 0 \\ \hline \end{array}$$

* عشان هشتغل على ال center بتاع الصورة اللي هو (z_5)

* لاحظ الفلتر غريب شوية بس كان كده أول ما طلع في 1965 باسم اللي عمله (Robert)

الناتج هيبقى بالشكل التالي (بنفذ ال Mask على المنطقة المحددة بالأخضر)

$$g_x = (z_9 - z_5) \quad \& \quad g_y = (z_8 - z_6)$$

وهنعتبر ال gradient image اللي هي $M(x,y)$ عند z_5 (يعني أصبح ال pixel)

$$M(x,y) = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{1/2}$$

OR more effeciently for computation

$$M(x,y) \simeq |z_9 - z_5| + |z_8 - z_6|$$

مع بعد كده تعديل على شغل Roberts وحول ال Mask بتاعك لحاجة 3x3 ؛ أسهل في الشغل من ال 2x2 بتاع Roberts ودول اللي هنتكلم عليهم من الأول

قبل ما ننقل عليه نشوف إزاي ال 1st derivative بمفهومها الأول (هنسميها look-ahead operators)

هنجيب g_x , g_y

احنا كنا عارفين المعادلة

$$\begin{array}{ccc} z_1 & z_2 & z_3 \\ z_4 & z_5 \xrightarrow{y+1} & z_6 \\ \downarrow{\scriptstyle x+1} & & \\ z_7 & z_8 & z_9 \end{array}$$

$$\boxed{\begin{aligned} g_x &= z_8 - z_5 \\ g_y &= z_6 - z_5 \end{aligned}}$$

وأحسب ال M بنفس الطريقة فوق

$$(1\text{-D}) \quad \frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\boxed{8} \quad (2\text{-D}) \quad \frac{\partial f}{\partial x} = f(x+1, y) - f(x, y)$$

جه بقى الـ Sobel operator عمل Mask 3x3 مش Roberts operators

original

$$\begin{matrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{matrix}$$

x

$$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

اللي تحت ناقص اللي فوق

y

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

اليمين ناقص الشمال

$$g_x = \underbrace{(z_7 + 2z_8 + z_9)}_{\text{اللي تحت}} - \underbrace{(z_1 + 2z_2 + z_3)}_{\text{اللي فوق}}$$

$$g_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$M(x,y) \simeq |g_x| - |g_y|$$

$$\simeq |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)|$$

$$+ |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

راجع بقى الـ Slide رقم [10.11] بعد الكلام ده ومعاها [10.10]

* تطبيق على الـ Sobel filtering في [10.12]

الصورة على شمال عدسات عيون و فيها مشكلة ، عمل عليها Sobel filter

جاب فيه صورة x و صورة y و عمل Magnitude ليهم جاب الصورة على اليمين عشان منها الظل في الزوايا وفيه الخطاء الدايرة

** Sobel filtering is good for automated inspection due to simplified computation task **

Combining Spatial enhancement methods

في سلايد [10.13] شارح تطبيق عشان يحسن صورة ، هيعديها على كذا عملية لحد ما يوصل للصورة h من الصورة a

* وبيوريك في نفس الفرق بين لو استخدمت Laplacian أو Sobel مثل

a	b
c	d

e	f
g	h

* هيعمل العمليات كلها بدون scaling وهيعمله في الآخر فاصل لما نخلص كل العمليات

* الخطوات اللي عملها (أرجع جدا اقرأ صفحة ١٩١ و ١٩٤ في المرجع)

① في البداية هو عايز يطلع details أكتر من الصورة

② عمل Laplacian operator طلعله الصورة (b) ، جمعها على (a) وطلع (c)

③ منه ملاحظ في (c) إنها noisy ، فعشان يتخلص من ال noise هيعمل smoothing بس بلاش ال median ، هيضيع ال features

④ فاللي تعال نعمل Sobel ونجيب (d) مفيهاش noise كتير وممكن نستعملها ك Mask للصورة (c) ، ال Sobel بيجيب منه ال gradient

⑤ هيعمل smoothing للصورة بتاعت ال gradient ويطلع (e) وبعد كده يضربها ك Mask في صورة ال Laplacian image فهيعمل $c * e$ ويطلع (f)

⑥ كده بقت ال edges عندي ظاهرة جدا ، أضيفها على الصورة الأصلية (a) وأطلع الصورة (g)

⑦ هيعلي ال Dynamic Range من خلال ال power Law أو ال Gamma Correction

ممكن تراجعها في سلايد (6.12) ، فعوض $\gamma = 0.5$ و $c = 1$ وجاب الصورة (h)

* كان ممكن يعمل الخطوة ⑦ بـ Histogram specification ، بس ال power Law أسرع في التطبيق ده تحديدا

* هكرر تاني ، أنصح بشدة المثال ده يتراجع من المرجع صفحة 191, 194